Integrating fast mobility in the OLSR routing protocol

Mounir Benzaid, Pascale Minet, Khaldoun Al Agha

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Integrating fast mobility in the OLSR routing protocol

Mounir Benzaid — Pascale Minet — Khaldoun Al Agha

N° 4510

Juin 2002

THÈME 1

ISSN 0249-6399 ISRN INRIA/RR--4510--FR+ENG

INRIA
ROCQUENCOURT

Integrating fast mobility in the OLSR routing protocol

Mounir Benzaid* , Pascale Minet† , Khaldoun Al Agha‡

Thème 1 — Réseaux et systèmes
Projet HIPERCOM

Rapport de recherche n° 4510 — Juin 2002 — 10 pages

Abstract: With the current increase in ad-hoc mobile networks in public domains (e.g. airports, parks, stations, etc.), and the widespread use of IEEE802.11 wireless LAN, there is a growing need to handle and manage fast mobility. Extending the coverage area of ad-hoc networks and taking into account fast mobility in routing protocols could offer a complementary solution to the UMTS for the fourth generation (4G) mobile networks. In this paper we present an extension of the Optimized Link State Routing protocol (OLSR), denoted Fast-OLSR. Fast-OLSR is designed to meet the need for fast mobility in Mobile Ad-hoc NETworks (MANETs). Performance evaluation of Fast-OLSR is done by simulation. Simulation results show that the loss rate can be minimized while maintaining a reasonable overhead traffic.

Key-words: mobile ad-hoc networks, wireless networks, fast mobility, routing protocol, OLSR.

* Email:*mounir.benzaid@inria.fr*
† Email:*pascale.minet@inria.fr*
‡ Email:*khaldoun.alagha@inria.fr*

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay Cedex (France)
Téléphone : +33 1 39 63 55 11 —Télécopie : +33 1 39 63 53 30

Intégration de la gestion de la mobilité rapide dans le protocole de routage OLSR

Résumé : Avec le déploiement des réseaux mobiles ad-hoc dans les lieux publics (par exemple, aéroports, parcs, gares, ...), et l'utilisation de plus en plus répandue des réseaux locaux sans fil IEEE802.11, il devient nécessaire d'offrir aux usagers une mobilité rapide. Un plus large déploiement des réseaux ad-hoc et la prise en compte de la mobilité rapide dans les protocoles de routage permettraient d'élaborer une solution complémentaire à l'UMTS en vue d'une intégration dans la 4^e génération des réseaux mobiles. Dans cet article, nous proposons une extension du protocole OLSR (Optimized Link State Routing), appelée Fast-OLSR. Cette extension Fast-OLSR est conçue dans l'objectif de gérer la mobilité rapide dans les réseaux ad-hoc mobiles (MANETs). Nous évaluons les performances de Fast-OLSR par simulation. Les résultats de simulation montrent que le taux de perte peut être réduit au minimum tout en maintenant un trafic de contrôle raisonnable.

Mots-clés : réseaux mobiles ad-hoc, réseaux sans fil, mobilité rapide, protocole de routage, OLSR.

1 Introduction

In response to the increasing popularity of mobile computers, Mobile IP [1] was developed to enable computers to maintain Internet connectivity while moving from one Internet attachment point to another. On the other hand, with recent technological advances in laptop computers and wireless data communication devices, wireless communications represent one of the fastest growing segments of the communications industry today. Wireless networks bring a new dimension to mobility; indeed they enable a user to access the Internet anywhere and at any time. At present, mobile wireless networks can be classified according to two main types: infrastructure and infrastructureless mobile networks; this last type of network is called an ad-hoc network [2, 3, 4].

A mobile ad-hoc network (MANET) [5] is a collection of mobile nodes that communicate using a wireless medium, forming an autonomous network. There is no centralized access point or pre-existing infrastructure. Such networks have dynamic, random, sometimes rapidly changing topologies, limited bandwidth, variable throughput links, and limited power (e.g. battery operated devices). When a node needs to communicate with another node, it uses either a direct wireless link or a multi-hop route to reach the destination. This means that all the nodes must incorporate routing capability to ensure that packets are delivered to the designated destination. Moreover ad-hoc routing protocols must minimize the induced control traffic. Several routing protocols are proposed in the MANET working group of IETF (Internet Engineering Task Force Internet). All these protocols generally deal with low mobility environment conditions.

With the increasing appearance of ad-hoc mobile networks in public domains (e.g. airports, parks, stations, etc.), there is an increasing need to handle and manage fast mobility. Low mobility can be considered to be the speed at which a pedestrian walks. On the other hand, fast mobility ranges from the speed of a cyclist up to that of a car travelling on highway. Extending the coverage area of ad-hoc networks and taking into account fast mobility in routing protocols could offer a complementary solution to the UMTS in fourth generation (4G) mobile networks. The design of a fast and efficient routing protocol is necessary to the performance of an ad-hoc network in particular in the case of large and dense networks with fast moving nodes. This paper focuses on how to deal with fast moving nodes in the OLSR routing protocol.

OLSR [2] is one of the protocols discussed in the MANET working group. The protocol, as described in this paper, inherits the stability of a link-state routing protocol and the availability of routes when needed due to its proactive nature. However, when a node is moving fast, the links with its neighbors are valid only during a short time interval. If packets are forwarded on an invalid link, not yet detected as broken, they are lost. Hence to minimize packet loss, broken links between the node and its neighbors must be quickly detected. In this paper, we propose the Fast-OLSR extension to account for fast nodes in routing while keeping the routing overhead as low as possible. This extension is based on the initial study [6]. A tradeoff is found between the routing overhead and the loss rate. As a result, the ad-hoc network nodes can move as fast as the cellular network nodes and the overhead remain acceptable.

The remainder of this paper is organized as follows: in Section 2, we describe the two families of routing protocols (i.e. reactive and proactive protocols) discussed in the MANET working group. In Section 3, we briefly present OLSR, a proactive protocol. In Section 4, we show how this protocol can be extended to take into account fast mobility. In Section 5, we evaluate the performance of the Fast-OLSR extension in terms of message loss and induced overhead. We first outline our simulation model and then analyse the simulation results of Fast-OLSR.

2 MANET Routing protocols

Different routing protocols are proposed in the MANET working group of the IETF. They address the problem of unicast routing, while taking into account the features of wireless, multi-hop, mobile ad-hoc networks. Such protocols can be divided into two categories: proactive and reactive, depending on the route discovery mechanism that is used.

With reactive protocols, a node discovers routes on-demand and maintains only active routes. Thus, a route is discovered whenever a source node needs to communicate with a destination node for which a route is not available. This discovery is based on pure flooding [7] in the network. The source node broadcasts a *route request* message to all its neighbors. The neighbors in turn rebroadcast the *route request* to their neighbors if they do not have a route to the destination. When the *route request* reaches either the destination or a node that has a valid route to the destination, a *route reply* message is generated and transmitted back to the source. This *route reply* follows the reverse path to the source; intermediate nodes, which receive the *route reply*, set up a route to the destination. Therefore as soon as the source receives the *route reply*, a route is created from the source to the destination. The advantage of reactive protocols is that no control message is needed for non-active routes. The drawback is the latency when establishing a route. Furthermore, the rapidly changing topology may break an active route and cause subsequent route discovery to establish a new route to a destination. Examples of reactive protocols include AODV [1] and DSR [8]. In AODV, based on "Distance Vector routing", the *route reply* contains the number of hops to reach the destination. In DSR, based on "Source Routing", the packet header carries the sequence of hops that the packet has to follow to reach a destination.

With proactive protocols, each node continuously maintains the routes to all other nodes in the network by periodic exchange of control messages. When a node needs to send a packet to any other node in the network, the route is immediately available. The main advantage of proactive protocols is that they do not introduce a delay before sending data. Furthermore, these types of protocols are useful for traffic patterns where a large subset of nodes is communicating with another large subset of nodes, and where the [source, destination] pairs are changing over time. As for fixed networks, two main approaches are used for proactive ad-hoc protocols: "Distance-Vector routing" and "Link State routing". Examples of proactive protocols include DSDV [9] (an adaptation of Routing Internet Protocol [10]), OLSR (an optimization of the Link State algorithm OSPF [11]) and TBRPF [12].

3 Optimized Link State Routing Protocol (OLSR)

OLSR [2] is a proactive routing protocol, providing the advantage of having routes immediately available in each node for all destinations in the network. It is an optimization of a pure link state routing protocol. This optimization is based on the concept of *multipoint relays* (*MPRs*) [13]. First, using *multipoint relays* reduces the size of the control messages: rather than declaring all links, a node declares only the set of links with its neighbors that are its "*multipoint relays*". The use of *MPRs* also minimizes flooding of control traffic. Indeed only *multipoint relays* forward control messages. This technique significantly reduces the number of retransmissions of broadcast control messages [14, 15].

OLSR is characterized by two types of control messages: neighborhood and topology messages, called respectively *Hello* messages and *Topology Control* (*TC*) messages. Indeed OLSR provides two main functionalities: the first is in Neighbor Discovery, and the second is Topology Disseminating. These will be detailed in the following.

3.1 Neighbor Discovery

Each node must detect the neighbor nodes with which it has a direct link. Due to the uncertainties in radio propagation, a link between neighboring nodes may enable the transmission of data in either one or both directions over the link. For this, each node periodically broadcasts *Hello* messages, containing the list of neighbors known to the node and their link status. The link status can be either *symmetric* (if communication is possible in both directions), *asymmetric* (if communication is only possible in one direction), *multipoint relay* (if the link is symmetric and the sender node of the *Hello* message has selected this node as a *multipoint relay*), or *lost* (if the link has been lost). The *Hello* messages are received by all one-hop neighbors, but are not forwarded. They are broadcast at a low frequency determined by the refreshing period "*Hello_interval*" (the default value is 2 seconds).

Thus, *Hello* messages enable each node to discover its one-hop neighbors, as well as it two-hop neighbors (the neighbors of its neighbors). This neighborhood and two-hop neighborhood information has an associated holding time "*Neighbor_hold_time*", after which it is no longer valid. On the basis of this information, each node can perform the selection of its *multipoint relays* as follows:

- ***MultiPoint Relay selection:*** Each node m of the network independently selects its own set of *multipoint relays* among its one-hop neighbors. The multipoint relay set of m, denoted $MPR(m)$ is computed in the following manner: it is a minimum subset of one-hop neighbors with a *symmetric* link, such that all two-hop neighbors of m have *symmetric* links with $MPR(m)$. This means that the *multipoint relays* cover (in terms of radio range) all the two-hop neighbors. Figure 1 shows the *multipoint relay* selection by node m. One possible algorithm for selecting these *MPRs* is described in [13]. The *multipoint relay* set is computed whenever a change in the one-hop neighborhood or two-hop neighborhood is detected, i.e. whenever a new one-hop or two-hop neighbor with *symmetric* link is detected, or a *symmetric* link with a one-hop or two-hop neighbor has failed.

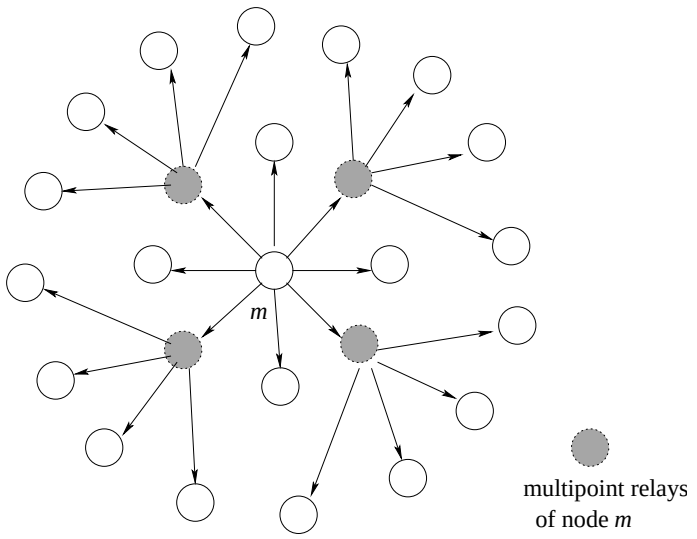

Figure 1: Multipoint relays of node m.

Each node m maintains the set of its "*multipoint relay selectors*" (*MPR selectors*). This set contains the nodes which have selected m as a *multipoint relay*. Node m only forwards broadcast messages received from one of its *MPR selectors*.

3.2 Topology Dissemination

Each node of the network maintains topological information about the network obtained by means of *TC* messages. Each node m selected as a *multipoint relay*, broadcasts a *TC* message at least every "*TC_interval*"(the default value is 6 seconds). If a change occurs in the *MPR selector* set, the next *TC* can be sent earlier (e.g. after some pre-specified minimum interval). The *TC* messages are flooded to all nodes in the network and take advantage of *MPRs* to reduce the number of retransmissions. The *TC* message originated from node m declares the *MPR selectors* of m. Thus, a node is reachable either directly or via its *MPRs*. This topological information collected in each node has an associated holding time "*Top_hold_time*", after which it is no longer valid.

The neighbor information and the topology information are refreshed periodically, and they enable each node to compute the routes to all known destinations. These routes are computed with Dijkstra's shortest path algorithm [7]. Hence, they are optimal as concerns the number of hops. Moreover, for any route, any intermediate node on this route is a *multipoint relay* of the next node. The routing table is computed whenever there is a change in neighborhood information or a change in topology information.

4 Fast-OLSR

In an OLSR ad-hoc network, when a node is moving fast, its neighborhood changes quickly and the default *Hello* frequency in OLSR is not sufficient to track the nodes motion. Therefore, routes to this node become inactive and messages sent to this node may be lost. A higher *Hello* frequency would overcome this problem, but at the cost of an additional control overhead.

We propose an extension of OLSR to address the issues at fast mobility nodes. This "Fast-OLSR" extension has two main objectives:

- The induced control traffic is tuned to node mobility, in such a way that it allows a fast node's motion to be tracked. I.e. there is very low overhead when there is no mobility and an appropriate overhead as mobility increases.
- The bandwidth must remain reasonable (i.e. maintained below a certain threshold).

Fast-OLSR has been designed as an extension of the OLSR protocol in so far as:

- Not all nodes are required to implement Fast-OLSR; OLSR and Fast-OLSR can coexist in the same mobile ad-hoc network.
- Fast-OLSR is based on the same basic principles as OLSR. It only differs in the Neighbor Discovery functionality, which is adapted to deal with fast mobility.

Moreover, to stay general and completely independent of the underlying link layer (e.g. IEEE 802.11, Bluetooth, etc.), Fast-OLSR does not make any assumption concerning the information provided by the chosen link layer. This means for instance that Fast-OLSR must work even when neither information on signal attenuation nor link layer notification of broken link are available. In such conditions, the only way that permits Fast-OLSR to detect the node mobility is to observe the neighborhood changes.

The basic idea of neighbor discovery in Fast-OLSR is to enable a fast moving node, m, to quickly discover a small number of neighbors. Among these a small number of *multipoint relays* are selected to maintain connectivity with other nodes in the network. To achieve this, a fast moving node establishes a small number of *symmetric* links refreshed at a high frequency by means of *Fast-Hellos*. Such links are called Fast links and this high frequency is determined by the refreshing period "*Fast_hello_interval*".

There are three main mechanisms in Fast-OLSR:

- Switching to the *Fast-Moving/Default* mode: when a node detects that it is moving fast (e.g. a high number of changes in its neighborhood), it switches to the *Fast-Moving* mode and starts sending *Fast-Hellos*. On the other hand, when a node in *Fast-Moving* mode detects that it is no longer moving fast (e.g. a small number of changes in its neighborhood), it switches back to the *Default* mode, which is the initial one.
- Establishing Fast links: a node in Fast-Moving mode sends *Fast-Hello* messages at high frequency. A *Fast-Hello* is similar to a *Hello*, however its size is smaller because it contains a reduced number of neighbor addresses. *Fast-Hello* messages are used to establish Fast links. When a node in *Default* mode receives a *Fast-Hello* from a node m in *Fast-Moving* Mode, it replies with a *Fast-Hello*. Among the received replies, node m selects a small number of *MPRs*. These are declared in the next *Fast-Hello*. "The declared nodes" will then broadcast *TC* messages to all the nodes in the network declaring that they are *MPRs* of m. Only nodes in *Default* mode can be selected as *MPRs*. This simplified selection of *MPRs* is used because the node m, which is moving fast, does not know its two-hop neighborhood.
- Refreshing Fast links and Detecting new/broken links: a node m in *Fast-Moving* mode sends *Fast-Hellos* containing the addresses of its *MPRs* and its *MPRs* reply with empty *Fast-Hellos*. Empty *Fast-Hello* save bandwidth usage, and they are sufficient to enable node m to know that it can always be reached by these *MPRs*. If an *MPR* of m has not received the *Fast-Hello* of m, it sends a *TC* message to inform all nodes in the network that it is no longer an *MPR* for m.

By means of *Fast-Hellos*, Fast-OLSR enables a broken link to be detected quickly. Hence the computation of a new route is made earlier, and message loss is reduced. Moreover, as the number of Fast links refreshed by a node in *Fast-Moving* mode is reduced, the additional overhead remains reasonable. As soon as a node in *Default* mode no longer has any Fast links, it stops sending *Fast-Hellos* and returns to the normal OLSR behavior, by sending only *Hello* and *TC* messages.

5 Simulation

In this section, we study the behavior of Fast-OLSR in worst-case conditions. We conduct a performance evaluation by simulation. The velocity of mobile nodes ranges from a cyclist's speed up to a car's speed on a highway. Two parameters are measured: the packet loss and the overhead produced by Fast-OLSR.

5.1 Simulation model

The considered simulation model is depicted in Figure 2. A fixed node C is located in a central position. It communicates by *symmetric* links with 6 nodes denoted m_1 to m_6. These nodes are fixed. One node m, moving with high speed, is also considered. It moves continually around m_1 to m_6 in a circular way. The velocity of m is constant, and its value depends on the simulation.

This model can, for instance, represent the effective architecture where C is a gateway and m_i, $i \in [1,6]$ are base stations that connect a fast moving wireless node m to another wired/wireless network via the gateway.

We consider that there is no overlapping between the radio coverage areas of two adjacent nodes m_i and m_{i+1}. This means that it is impossible for m to be in a soft handoff situation where it can receive or send packets from both m_i and m_{i+1}. Thus, node m loses the connectivity with m_i before having connectivity with m_{i+1}. We make this assumption in order to study the behavior

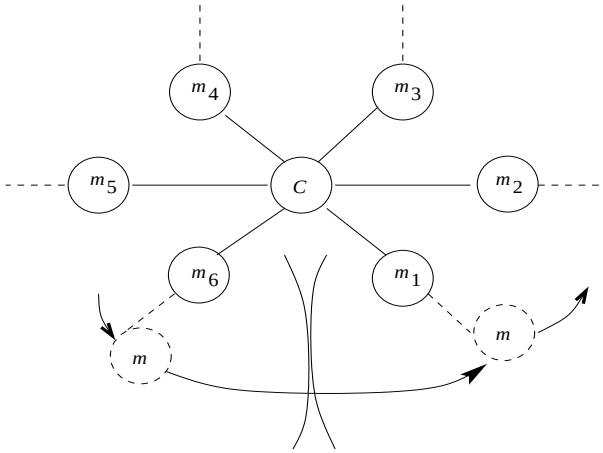

Figure 2: Simulation model.

of Fast-OLSR in worst-case conditions. Indeed, overlapping areas would enable a mobile node to maintain connectivity while a new route is being established. Moreover, we assume that neither link layer notification, nor information on signal attenuation are available. No link hysteresis is assumed to be provided.

In each simulation, there is an initialization time in which there is an exchange of *Hello* and TC messages to establish the links between node C and nodes m_1 to m_6. Then, m switches to *Fast-Moving* mode. Fast-OLSR performances are evaluated while m is moving 9 times around m_1 to m_6.

The signal attenuation distance is fixed to 60 meters (mean value of the attenuation distance in *IEEE802.11*). This model can be applied with *IEEE802.11*. In such a case, collisions would occur, they result from simultaneous transmissions of C and m_i. As in all considered scenarios, they are reduced to a minimum, and they are not considered in our model.

5.2 Simulation Measurements and Parameters

Let us assume that a Constant Bit Rate stream of packets is being transmitted from node C to mobile m. With our model, C is always two hops from m but the next hop to reach m changes when m moves. We are interested in evaluating the number of lost packets due to hard handoffs, assuming the worst conditions (no buffering and no retransmission).

- **Packet Loss:** This number is computed as follows. We first compute the number of lost packets during a handoff between nodes m_i and m_{i+1}. Packets are lost during the time interval $[t_1, t_2\ )$ where t_1 is the time when m loses connectivity with m_i and t_2 is the time when m can be reached through m_{i+1}. The total number of lost packets is obtained by adding the number of packets lost in each handoff generated during the simulation.

- **Induced overhead:** We compute the overhead induced by Fast-OLSR due to fast mobility of node m. This overhead is equal to the total number of (i) *Fast-Hello* messages generated in the network and (ii) the TC messages sent each time a fast link is established or broken.

- **Parameters:** We have measured the packet loss and the overhead induced in various simulations, each simulation being characterized by two parameters: first, the refreshing period of fast links (i.e., *Fast_hello_interval*), secondly the velocity of m in kilometers per hour.

5.3 Simulation results

Figures 3 and 4 highlight the performance of Fast-OLSR in terms of loss rate and overhead, when node C sends CBR traffic to mobile m. Mobility ranges from 20 km/h to 150 km/h.

Figure 3 depicts the packet loss rate versus the mobility. Several curves are drawn with regard to the value of the *Fast_hello_interval* (from 100 ms to 300 ms). As Figure 3 shows, the greater the *Fast_hello_interval* is, the greater the loss rate is, and, of course, the packet loss becomes greater as the speed increases.

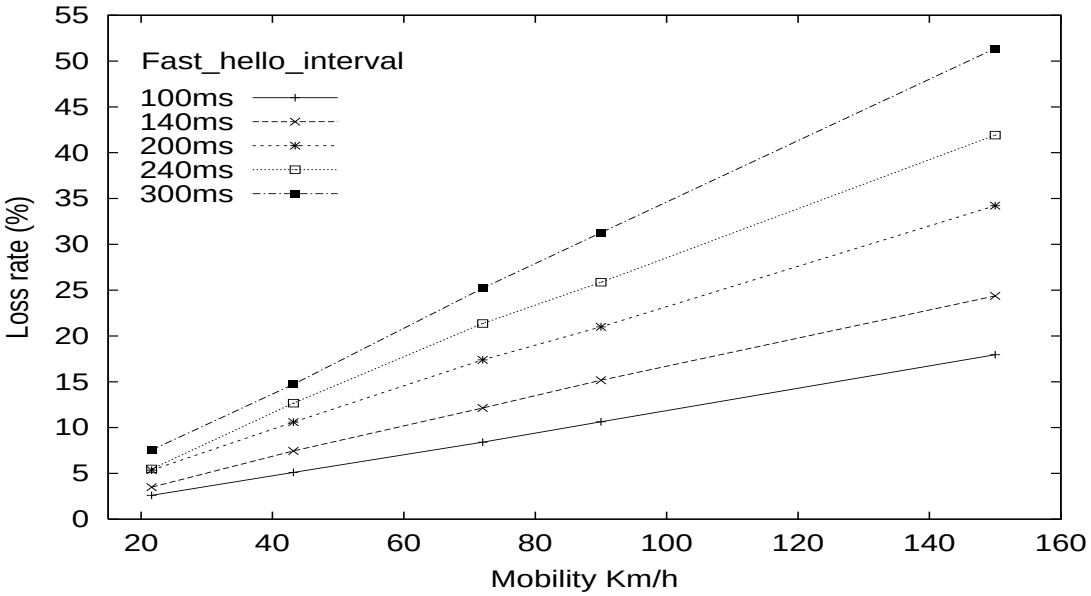

Figure 3: Loss rate versus mobility.

Our simulation results show that the packet loss rate can be maintained smaller than 10.6% for a velocity up to 90 km/h. Recall that these results are obtained in a worst-case where is no overlapping radio coverage area of two adjacent nodes m_i, m_{i+1}(see Figure 2) and no buffering or retransmission are considered. The routing overhead produced for the smallest *Fast_hello_interval* (i.e. 100ms) is kept below 7.7 kbit/s. To establish a comparison, this interval is equivalent to the 120 ms *Slow Associated Control CHannel* (*SACCH*) in GSM radio interface. The *SACCH* channel is used to report received signal strength and thus triggers the handoff between cells.

For a given maximum acceptable loss rate and the maximum reachable speed, we can determine the largest *Fast_hello_interval* that produces the least routing overhead. First we draw the vertical line indicating the maximum speed, and the horizontal line representing the acceptable loss rate. The acceptable region is then delimited by the x-axis, the y-axis, the speed line and the loss rate line. The most appropriate *Fast_hello_interval* is the one given by the intersection of the highest curve in the acceptable region with the speed line. For example, for a maximum speed of 125 km/h and a loss rate of 15%, the only possible *Fast_hello_interval* is 100 ms. However, for a speed limited to 40 km/h with a loss rate of 10%, there are several possibilities for the *Fast_hello_interval*: 100 ms, 140 ms, 200 ms. We select the highest curve, which gives us a *Fast_hello_interval* of 200 ms. Indeed this value incurs the smallest overhead in the acceptable region.

Figure 4 depicts the routing overhead induced by Fast-OLSR versus the mobility. Several curves are drawn with regard to the value of the *Fast_hello_interval* (from 100 ms to 300 ms). As Figure 4 shows, the smaller the *Fast_hello_interval* is, the greater the routing overhead is. The routing overhead becomes slightly greater as the speed increases.

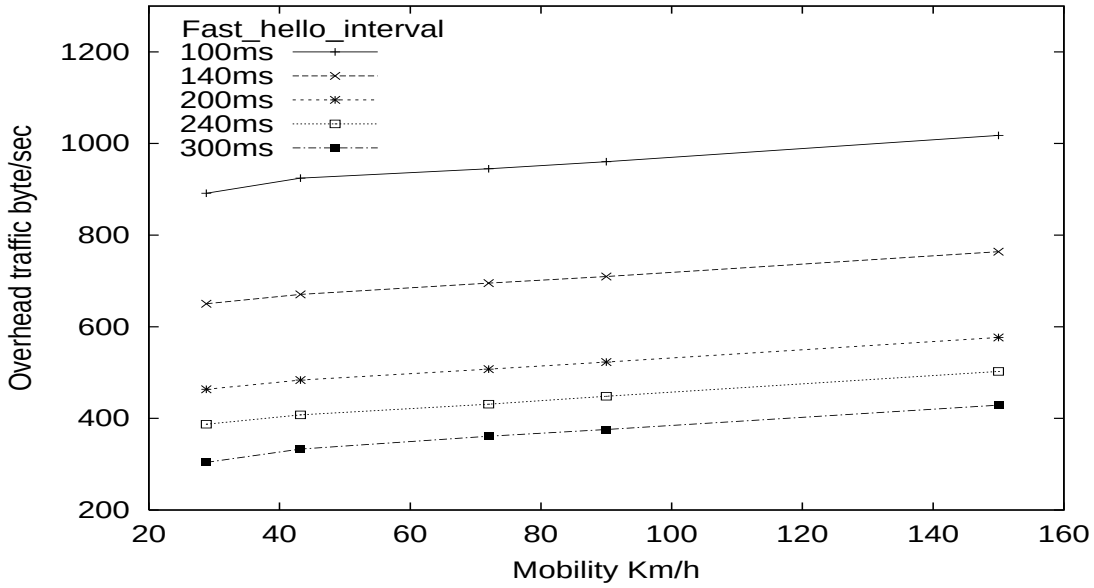

Figure 4: Overhead traffic versus mobility.

Figure 4 also shows the additional overhead resulting from a decrease in the *Fast_hello_interval*. The choice of a smaller *Fast_hello_interval* is justified only when the resulting gain in loss rate is significant. For example, for a maximum speed of 50 km/h and a loss rate of 10%, a *Fast_hello_interval* of 140 ms is chosen. A *Fast_hello_interval* of 100 ms would improve the loss rate (it would be 7%) but at the cost of an additional overhead of 1 kByte/s. On the other hand, for a maximum speed of 90 km/h and a loss rate of 25%, a *Fast_hello_interval* of 240 ms is chosen. It improves the loss rate (it would be 30% with a *Fast_hello_interval* of 300 ms) at the cost of an additional overhead of only 30 Byte/s.

Results of our simulations are summarized in Table 1. Four classes of mobile node are defined. Each class is defined by its maximum speed. For each class, we determine the best *Fast_hello_interval* and associated maximum routing overhead in terms of kbit per second, compliant with the acceptable loss rate.

Table 1. Mobility classes and associated results.

Class	Speed	Loss rate $\leq 10\%$		$10\% <$ Loss rate $\leq 15\%$		$15\% <$ Loss rate $\leq 20\%$	
	km per hour	Fast-hello ms	Overhead kbps	Fast-hello ms	Overhead kbps	Fast-hello ms	Overhead kbps
Cyclist	$20 \leq S < 45$	140	5.4	240	3.7	>300	2.7
Urban	$45 \leq S < 90$	100	7.7	140	5.7	140	5.7
Road	$90 \leq S < 120$	×	×	100	7.8	140	6.2
Highway	$120 \leq S < 150$	×	×	×	×	100	8.1

6 Conclusion

With the deployment of mobile ad-hoc networks in public domains (e.g. highways, cities, etc.), routing protocols must support fast mobility. In this paper, we have proposed a Fast-OLSR, an extension of OLSR dealing with fast mobility. Fast-OLSR maintains connectivity with fast moving nodes, while maintaining a reasonable overhead. The performances of Fast-OLSR are evaluated by simulation in worst-case conditions (no buffering, no retransmission, no overlap in coverage areas and no link layer notification). Simulation results show how to tune the value of the refreshing period (*Fast_hello_interval*) for different classes of mobile and different acceptable loss rates. Thus, ad-hoc network nodes can move as fast as cellular network nodes with a velocity that can reach 150 km/h and the loss rate is maintained below 15% with an acceptable overhead (the refreshing period being equal to 100 ms).

As further work, we will study how to dynamically adjust the value of the refreshing period as a function of the mobile speed and the acceptable loss rate.

References

[1] C. Perkins, "IP mobility support for IPv4," RFC 3220, IETF: The Internet Engineering Task Force, http://www.ietf.org, January 2002.

[2] P. Jacquet, P. Muhlethaler, A. Qayyum, A. Laouiti, L. Viennot, T. Clausen, "Optimized link state routing protocol," draft-ietf-manet-olsr-04.txt, IETF: The Internet Engineering Task Force, http://www.ietf.org, March 2001.

[3] J. Broch, D. Maltz, D. Johnson, Y. Hu, J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *ACM Mobicom'98*, (Dallas, USA), October 1998.

[4] C. Perkins, *Ad Hoc Networking*. Addison Wesley, 2000.

[5] Mobile Ad-hoc Networks (MANET), "manet-charter," tech. rep., IETF: The Internet Engineering Task Force, http://www.ietf.org/html.charters/manet-charter.html, 2002.

[6] P. Jacquet, P. Muhlethaler, P. Minet, A. Qayyum, A. Laouiti, L. Viennot, T. Clausen, "OLSR extensions," draft-ietf-manet-olsr-extensions-00.txt, IETF: The Internet Engineering Task Force, http://www.ietf.org, August 2001.

[7] A. S. Tanenbaum, *Computer Networks*. Prentice Hall, 1996.

[8] J. Broch, D. Johnson, D. Maltz, "The dynamic source routing protocol for mobile ad hoc networks," draft-ietf-manet-dsr-01.txt, IETF: The Internet Engineering Task Force, http://www.ietf.org, December 1998.

[9] C. Perkins, P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *ACM SIGCOMM'94*, (London, UK), August 1994.

[10] J. Malkin, "Routing information protocol (RIP) version 2," RFC 1388, IETF: The Internet Engineering Task Force, http://www.ietf.org, January 1993.

[11] J. Moy, "Open shortest path first (OSPF) version 2," RFC 2328, IETF: The Internet Engineering Task Force, http://www.ietf.org, January 1998.

[12] R. G. Ogier, F. L. Templin, B. Bellur, M. G. Lewis, "Topology broadcast based on reverse-path forwarding (TBRPF)," draft-ietf-manet-tbrpf-05.txt, IETF: The Internet Engineering Task Force, http://www.ietf.org, March 2002.

[13] A. Qayyum, L. Viennot, A. Laouiti, "Multipoint relaying: An efficient technique for flooding in mobile wireless networks," Research Report RR-3898, INRIA, http://www.inria.fr, February 2000.

[14] P. Jacquet, P. Muhletaler, A. Qayyum, A. Laouiti, T. Clausen, L. Viennot, "Optimized link state routing protocol," in *IEEE INMIC*, (Pakistan), December 2001.

[15] A. Qayyum, A. Laouiti, L. Viennot, "Multipoint relaying technique for flooding broadcast messages in mobile wireless networks," in *HICSS: HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES*, (Hawai, USA), January 2002.

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot-St-Martin (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399